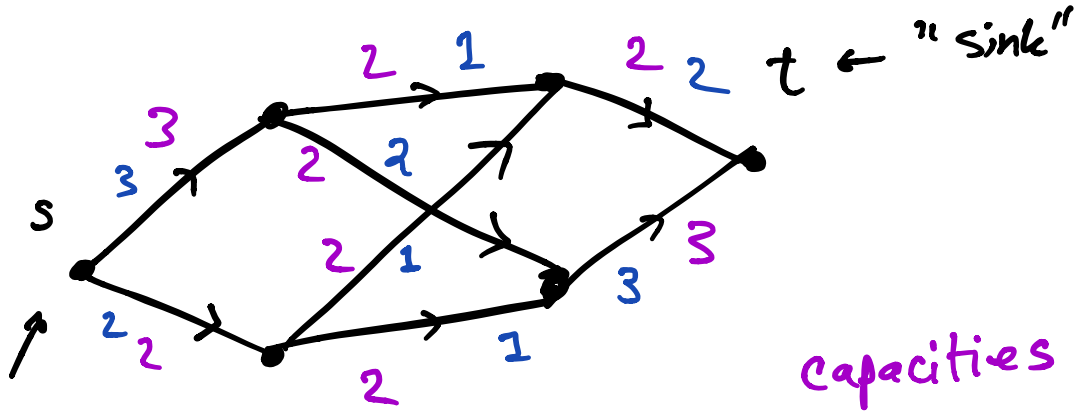
Lecture 12

Plan: 1) Def, examples
2) Min-flow, max-cut

Next time: Fast Algorithms for flows

Example: water pipes



- Except at source & sink, water going in must go out.
- flow cannot exceed capacity.

Question: how much water can be sent through? Above, 5 units.

- Obviously can also model electricity, traffic, etc.

more formally:

Def (Flow Network)

- $G = (V, E)$ directed graph.
- Two special vertices $s, t \in V$. "source" "sink"
- $u : E \to \mathbb{R}$ "upper capacity".
- $\ell : E \to \mathbb{R}$ "lower capacity". (if omitted, default is $\ell = 0$).

- A **flow** is function $x : E \to \mathbb{R}$

satisfying

$$\ell(e) \le x_e \le u(e) \quad \forall e \in E$$

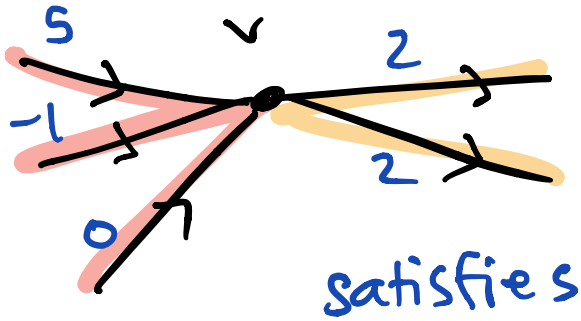
and "**flow conservation**"

$$(*) \sum_{e \in \delta^+(v)} x_e - \sum_{e \in \delta^-(v)} x_e = 0 \quad \forall v \in V \setminus \{s,t\}.$$

$\delta^+(v)$: set of edges leaving v.

$\delta^-(v)$: set of edges entering v.

E.g.



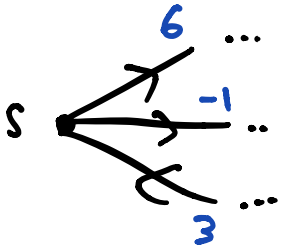
$\delta^-(v)$ $\delta^+(v)$

satisfies $(*)$.

- **value** of flow x is

$$|x| := \sum_{e \in \delta^+(s)} x_e - \sum_{e \in \delta^-(s)} x_e.$$

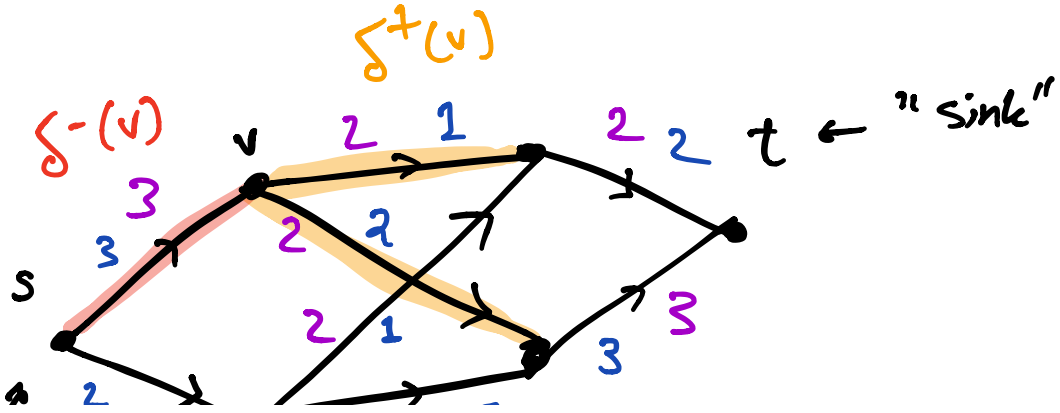
e.g.



value $6 - 1 - 3 = 2$.

- **maximum flow problem:**
  find flow x w/ largest value $|x|$.

E.g. (flow from earlier).



"source"

flow x has value 5

(upper) capacity u

lower $l = 0$.

x is maximum!

Notes

- $|x|$ = amt. leaving source

can also be expressed in terms of sink

$|x|$ = amt. entering sink

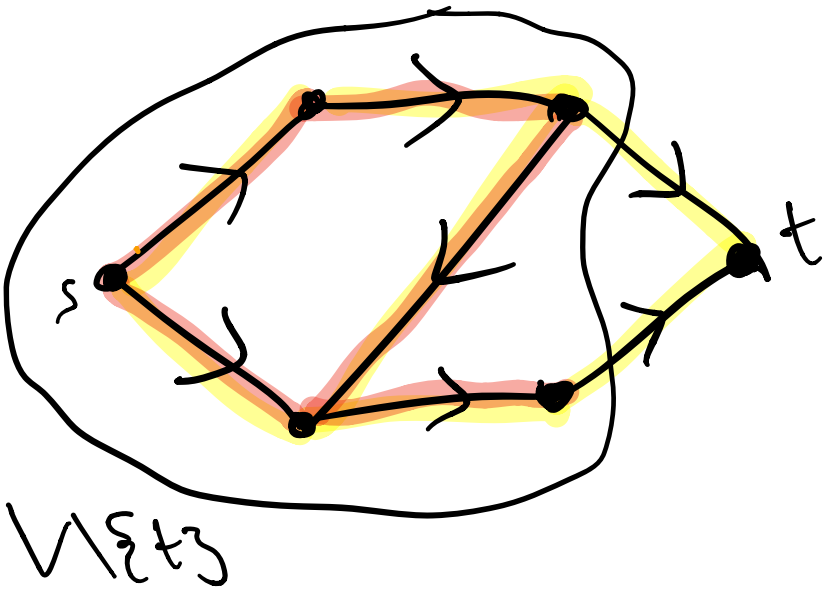
$$= \sum_{e \in \delta^-(t)} x_e - \sum_{e \in \delta^+(t)} x_e$$

Pf: by flow-cons.

$$|x| \overset{!}{=} \sum_{u \in V \setminus t} \left(\sum_{e \in \delta^+(u)} x_e - \sum_{e \in \delta^-(u)} x_e \right)$$

$$= \sum_{e \in \delta^-(t)} x_e - \sum_{e \in \delta^+(t)} x_e$$

all x_e except $e \in \delta(t)$ cancel



- max flow problem is an LP.

$$\max \sum_{e \in \delta^+(s)} x_e - \sum_{e \in \delta^-(s)} x_e$$

subject to

$$\sum_{e \in \delta^+(v)} x_e - \sum_{e \in \delta^-(v)} x_e = 0 \quad \forall v \in V \setminus \{s,t\}$$

$\sum_{e \in \delta^+(v)} x_e - \sum_{e \in \delta^-(v)} x_e = 0 \quad \forall v \in V \setminus \{s,t\}$

and $\ell(e) \le x_e \le u(e) \quad \forall e \in E.$

Theorem If ℓ, u integral, is integral max flow! * (IP=LP).

* if problem feasible.

Proof: Total unimodularity!

Express constraints in matrix form:

flow cons

$$\max\{c^T x : Nx = 0\}$$

capacity

$$\begin{cases} Ix \le u \\ x \ge \ell \end{cases}$$

So matrix is

$$A = \begin{bmatrix} N \\ \hline I \end{bmatrix}$$

why sufficient that A TU?

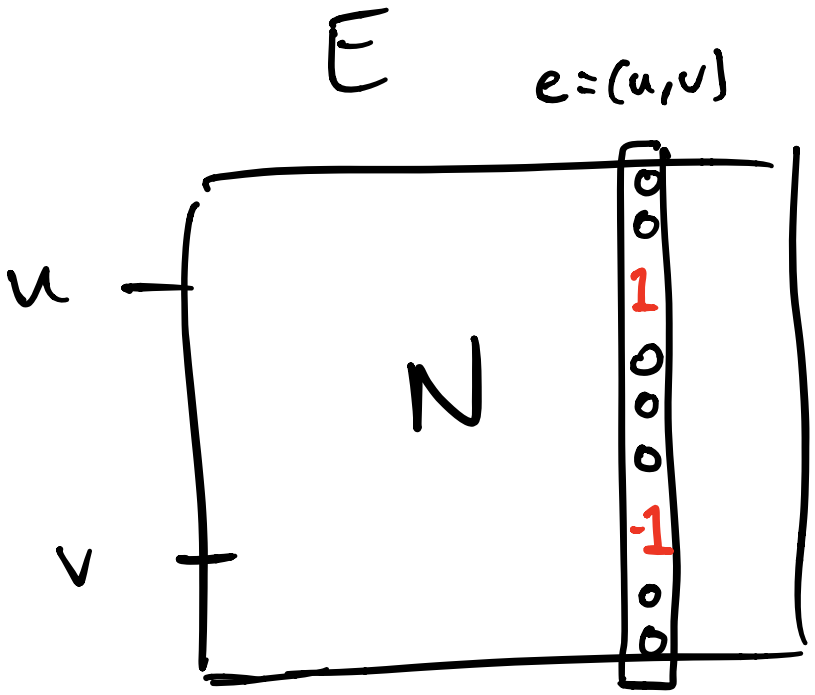
Exercise:

$$\{x : Ax \,\Delta\, b,\ x \geq\}$$

is integral if A is TU whenever Δ has $\geq, \leq, =$ and and ℓ integral.

Showing A is TU:

- Enough to show N is TU.
  (just expand down rows of I that appear).
- What is N?



N is <u>transpose</u> of directed incidence matrix!

1.K.

$$N_{ue} = \begin{cases} 1 & \text{if } e \in \delta^+(u) \\ -1 & \text{if } e \in \delta^-(u). \end{cases}$$

- Directed incidence matrices (and their transposes) are TU: Quiz extra credit :)

3 Cases for submatrix M of N:

(i) Col of all 0: $\det M = 0$ ✓

(ii) Col of one ±1: expand down it, get smaller submatrix. ✓

(iii) Every column has one +1, one −1, rest 0's.

Sum of rows is 0!

$\mathbb{1}^T M = 0 \Rightarrow \det M = 0$. ✓

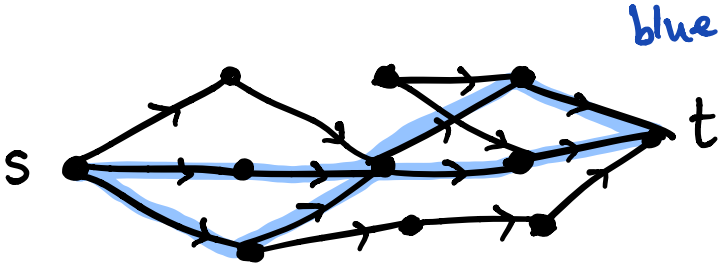
Could also use discrepancy. □

Special cases:

1) Edge-disjoint paths:

Setting $\ell = 0$, $u = \mathbb{1}$,
max flow is max # edge-disjoint s-t paths in G.

Why? Know is integer max flow; must be 0-1.



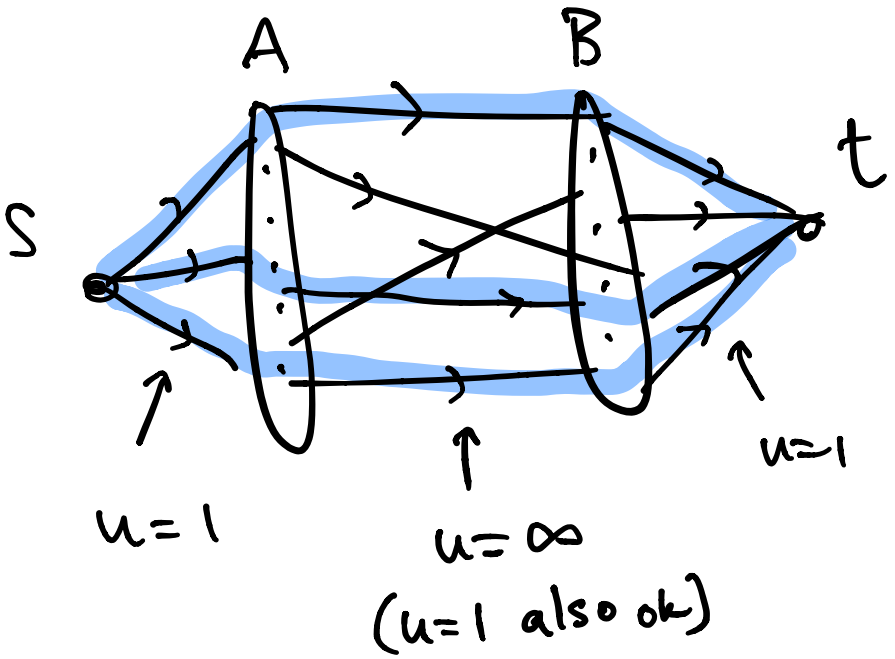
Perform flow decomposition:

Remove paths 1 at a time.

2) Bipartite matchings

$H = (A, B, E)$ bipartite graph

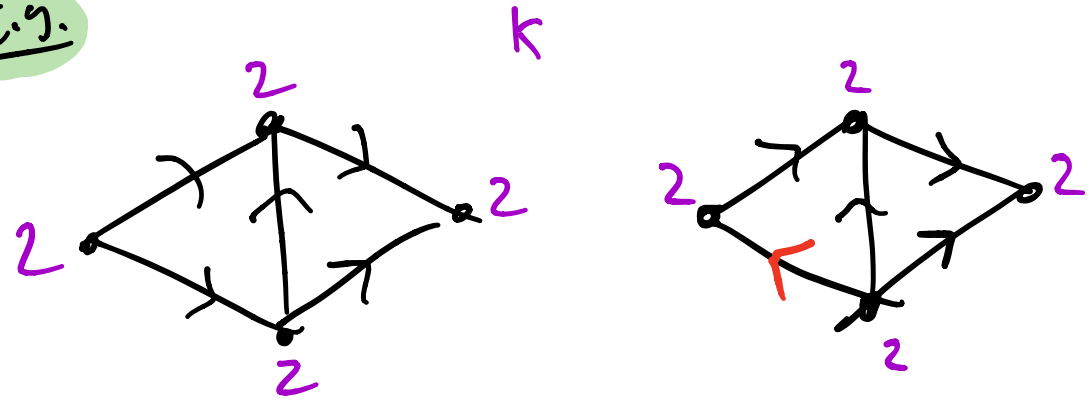
Direct edges across, add vertices s,t.



integer flow x $\longleftrightarrow$ matching of size $|X|$.

3) Orientations: Given directed graph G, orient edges so indegree of each vertex $v \in k(V)$.

E.g.



Ex: Figure out flow network.

The dual of this LP leads to the notion of cuts.

"cuts obstruct flows"

S-t cuts

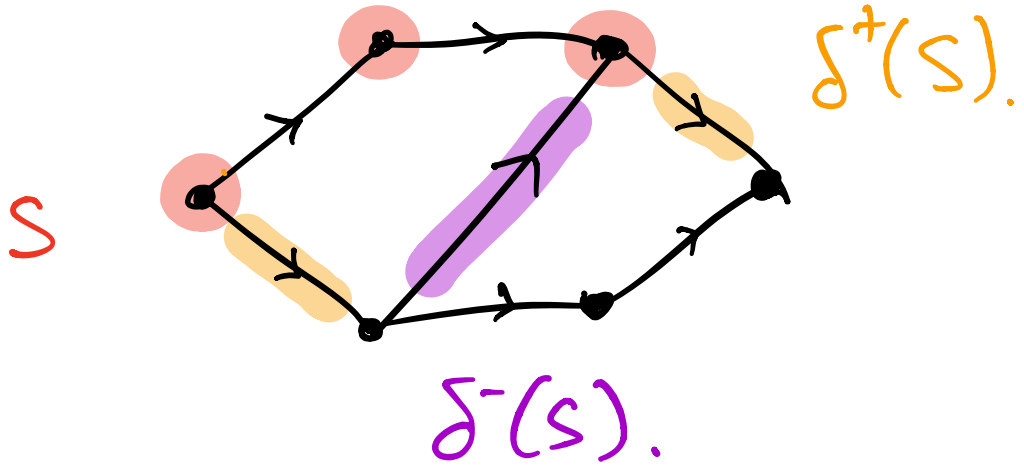
- Given digraph $G=(V,E)$, $S \subseteq V$,

  cut is set of arcs leaving S:

$$\delta^+(S) = \{(u,v) \in E : u \in S, v \in V \setminus S\}.$$

$$\delta^-(S) := \delta^+(V \setminus S).$$

E.g.



- Abuses of notation:
  - (i) For $v \in V$, $\delta(v) := \delta(\{v\})$
  - (ii) Conflate S, $\delta^+(S)$.

- S is **s-t cut** if $s \in S, t \notin S$.

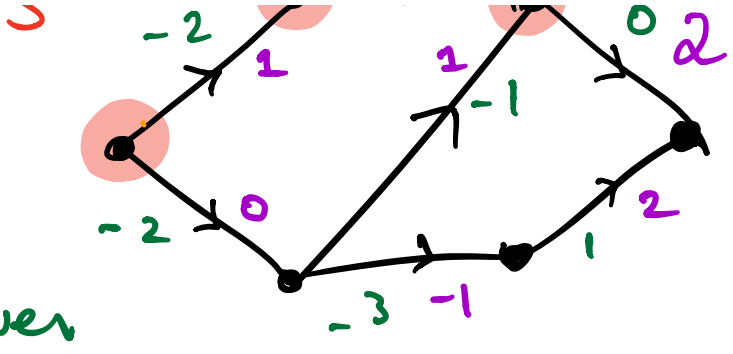
How do cuts bound flows?

- Given flow network $G = (V, E)$, capacities u, l,

capacity of cut S

$$c(S) = \sum_{e \in \delta^+(S)} u(e) - \sum_{e \in \delta^-(S)} l(e)$$

e.g.



lower

$$c(S) = 0 + 2 - (-1) = 3$$

- Note: $\forall$ flow x, capacities $\Rightarrow$

$$c(S) \geq \underbrace{\sum_{e \in \delta^+(S)} x_e - \sum_{e \in \delta^-(S)} x_e}_{*}.$$

- If S is s-t cut,

$$* = |x|!$$

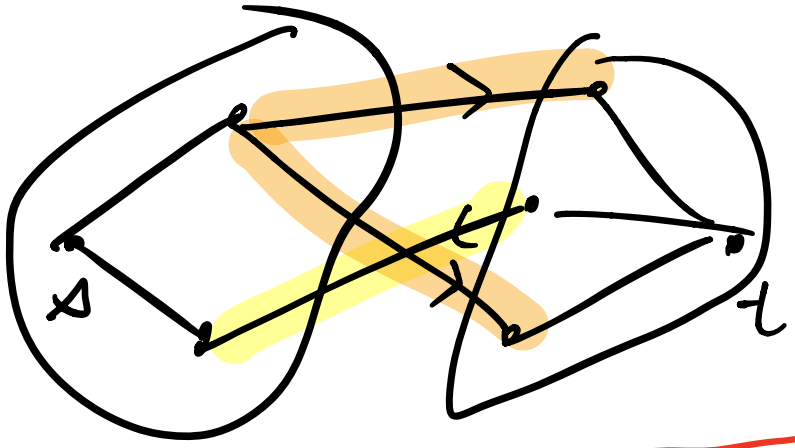
why? similar argument

to flow out of S = into $\bar{S}$:

$$|x| = \sum_{\delta^+(S)} x_e - \sum_{\delta^-(S)} x_e$$

$$= \sum_{v \in S} \left(\sum_{\delta^+(v)} x_e - \sum_{\delta^-(v)} x_e \right)$$

cancels except $\delta^+(S)$, $\delta^-(S)$.



- Thus **weak duality**

$$\max_{\text{flows } x} |x| \leq \min_{\substack{s\text{-}t \\ \text{cuts } S}} C(S)$$

- if flow instance feasible, strong duality holds!

Theorem (max-flow, min-cut)

For any feasible flow network,

$$\max_{\text{flows } x} |x| = \min_{\substack{s\text{-}t \\ \text{cuts } S}} c(S)$$

Proof: Could use LP duality, TU.

Today: "Primal-Dual";
Develop algorithm to find flow, show at termination find matching cut.

Algorithm (Augmenting flows)

- Assume begin w/ feasible flow x.
  e.g. if $\ell \le 0$, $u \ge 0$ $x = 0$ suffices.

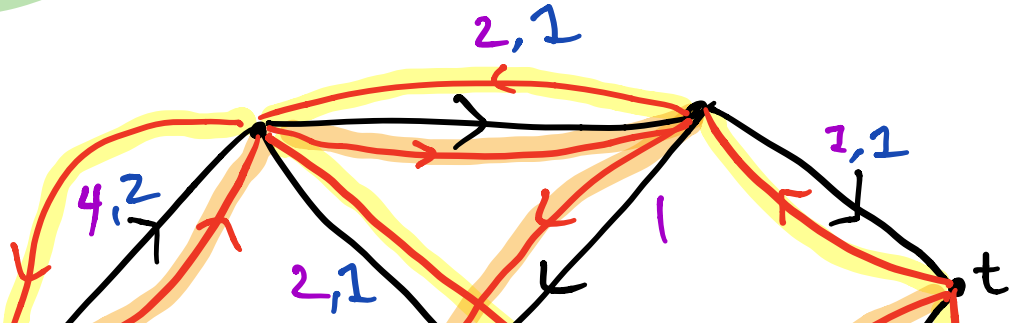
Repeat until termination:

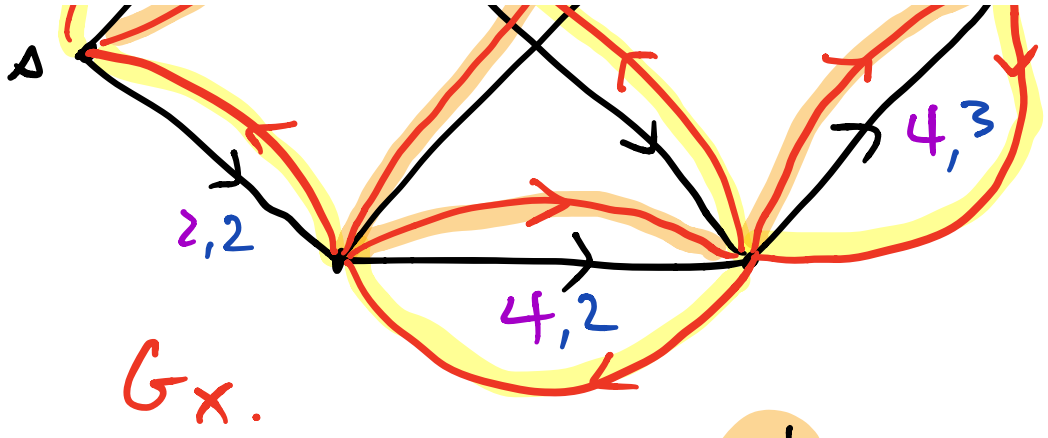
- Define residual graph $G_x = (V, E_x)$

(a) $(i,j) \in E_x$ if $(i,j) \in E$, $x_e < u(e)$
"could increase"

(b) $(i,j) \in E_x$ if $(j,i) \in E$, $x_e > \ell(e)$.
"could decrease"

e.g. $\ell = 0$, u, x





G_x.

(a) are "forward" arcs, P^+

(b) "backward arcs P^-

Case (i):

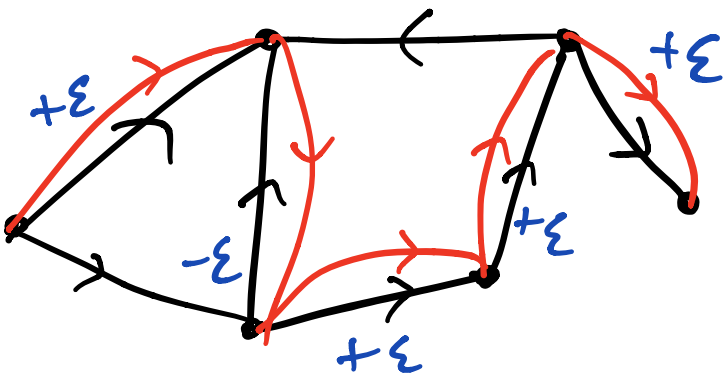
If $\exists$ directed s-t path P in G_x:

- "augment x along P":

$$x'_e = \begin{cases} x_e + \epsilon & e \in P^+ \\ x_e - \epsilon & e \in P^- \\ x_e & e \notin P. \end{cases}$$

- Observe x' satisfies flow conservation:

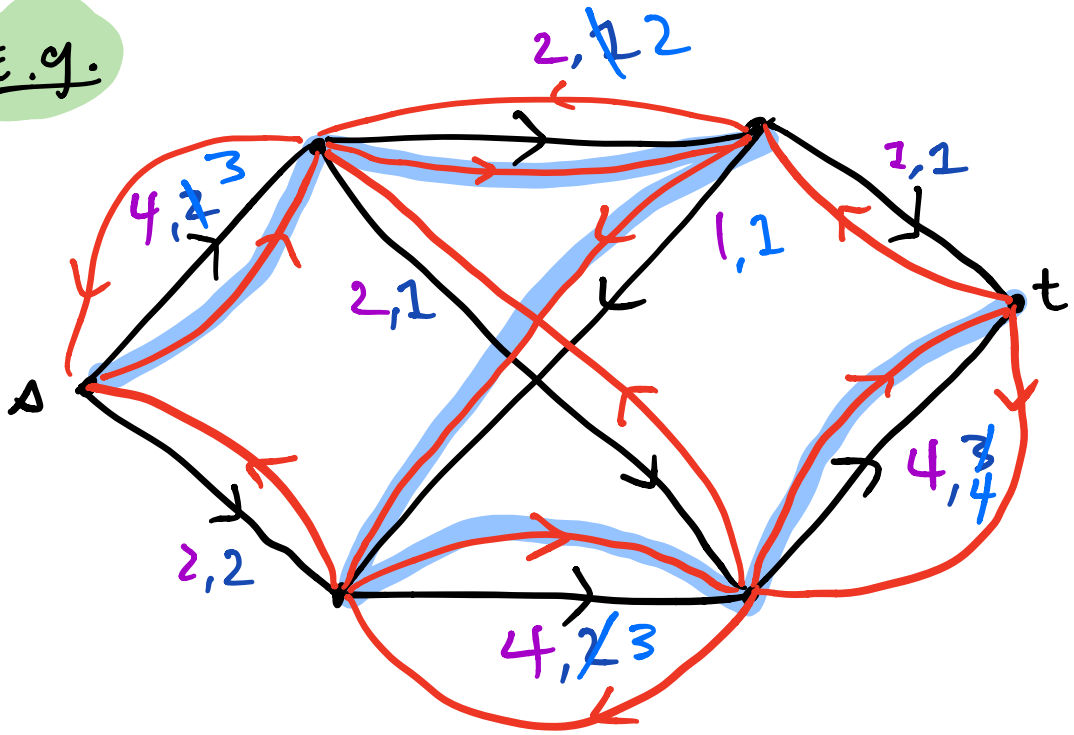
e.g.



- For x' to be feasible, choose

$$\epsilon = \left\{ \min_{e \in P^+} u(e) - x_e, \quad \min_{e \in P^-} x_e - \ell(e) \right\}$$

E.g.



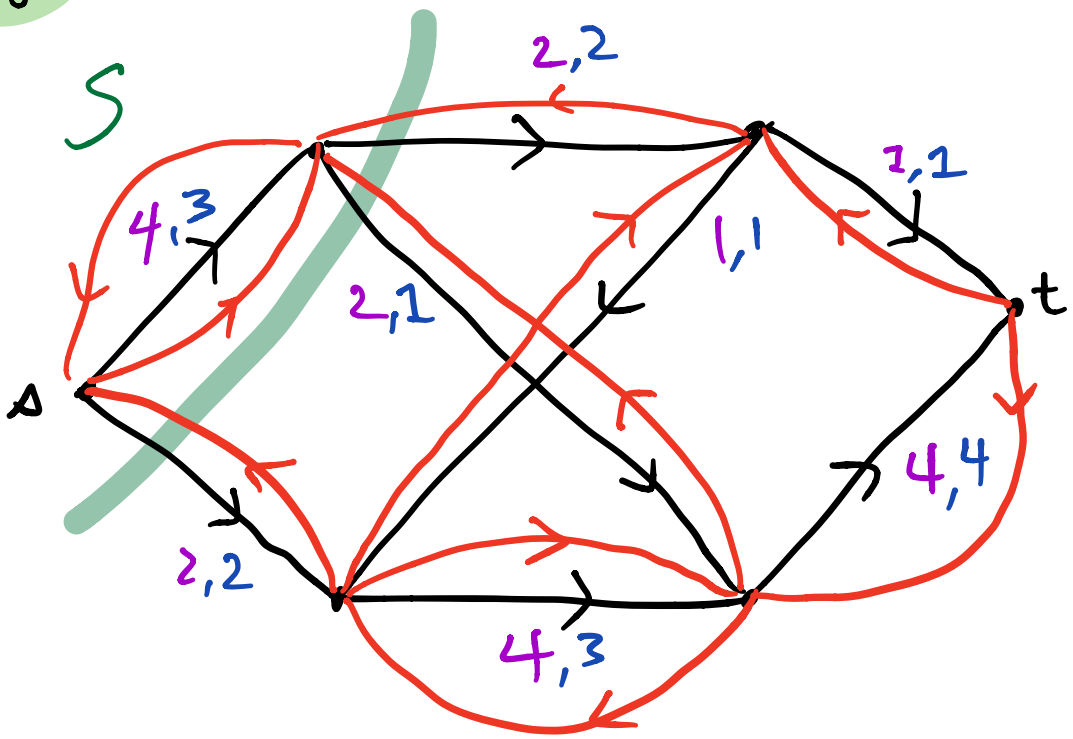
- we have $|x'| = |x| + \epsilon$.
- Set $x \leftarrow x'$.

Case (ii): No directed s-t path in G_x.

- Let $S = \{$vertices reachable from s in $G_x\}$

- By assumption, S is s-t cut.

E.g.



- Claim: $C(S) = |X|$.

why? if $e \in \delta^+(S)$, $x_e = u(e)$.

(else endpt. reachable).

similarly, if $e \in \delta^-(S)$, $x_e = \ell(e)$.

Means

$$c(S) = \sum_{e \in \delta^+(S)} u(e) - \sum_{e \in \delta^-(S)} \ell(e)$$

$$= \sum_{e \in \delta(S)} x_e - \sum_{e \in \delta^+(S)} x_e = |X|.$$

- Terminate, output X, S.

Does this prove theorem?

Not quite!

What if algorithm never terminates?

- if rational, ok b/c can assume integral & always increase flow by ≥ 1.

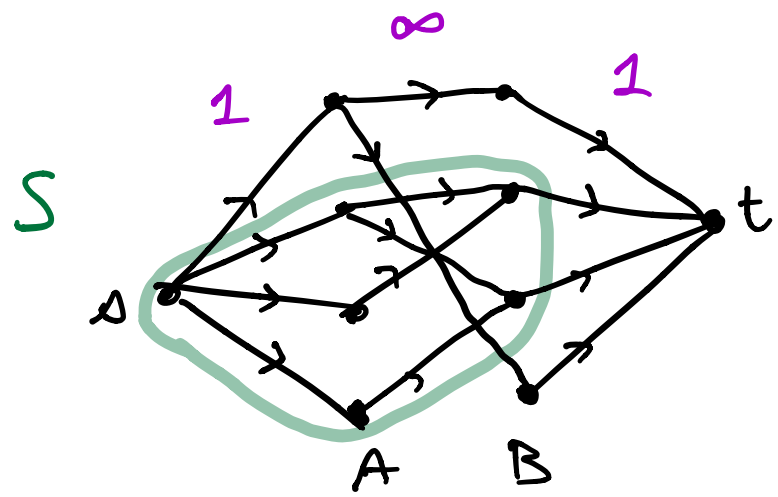
- if capacities irrational, can run forever (even for $|V| = 6$!)
- However, we can fix this:

  Because a max flow x exists (max flow is just an L.P. afterall) just start alg. with max flow x, must terminate immediately. (else value would increase). □

Applications of Max-flow Min-cut:

1) König's Theorem:

- Recall flow network capturing bipartite matching.



- A min cut cannot contain any of the ∞ edges; thus is in original graph.

$$C = (A \setminus S) \cup (B \cap S)$$

is vertex cover;

$$c(S) = |C| \; !$$

- Thus max matching = min vertex cover; another way to show König's!

2) Edge-disjoint paths

Menger's Theorem:

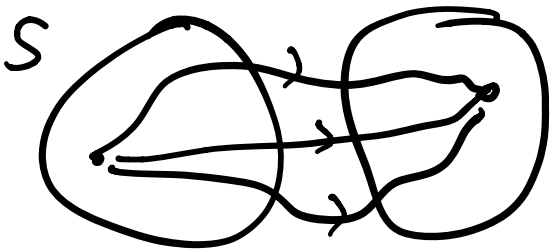
In directed graph $G=(V,E)$,

$\exists$ k edge-disjoint s-t paths

$\Leftrightarrow$

for all $S \subseteq V \setminus \{t\}$, $s \in S$,

$$|\delta^+(S)| \geq k.$$



Next time:

efficiently finding flows.